Trustworthy Machine Learning and Robust Artificial Intelligence

Tom Dietterich
Distinguished Professor (Emeritus)
Oregon State University

Chief Scientist, BigML
Corvallis, Oregon, USA

Outline

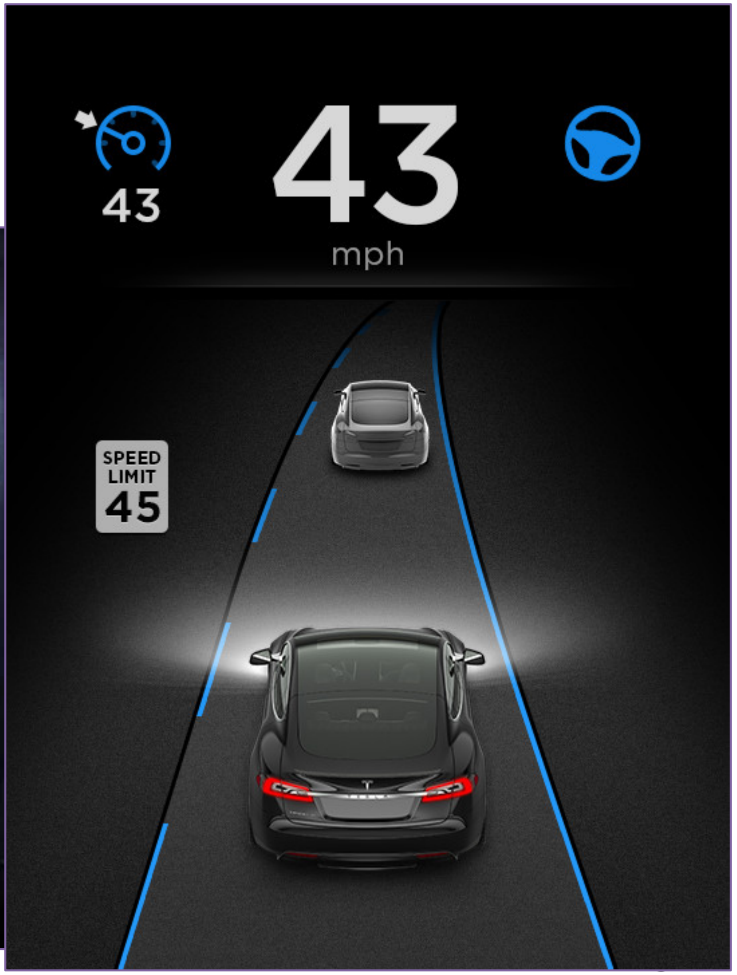
- Motivation: Why Trustworthy and Robust AI?
- Part 1: Trustworthy Machine Learning
- Part 2: Robust Artificial Intelligence
    - Part 2A: Robust Autonomous AI Systems
    - Part 2B: Robust Human/AI Teams

Self-Driving Cars

Credit: The Verge

Credit: delphi.com

Tesla AutoSteer



Credit: Tesla Motors

Automated Stock Trading

WIRED

CADE METZ BUSINESS 01.25.16 7:00 AM

THE RISE OF THE ARTIFICIALLY INTELLIGENT HEDGE FUND

The Economist

Oct 3rd 2019

The rise of the financial machines

Forget Gordon Gekko. Computers increasingly call the shots in financial markets

Luca D'Urbino

Autonomous Weapons

Northroop Grumman X-47B

Credit: Wikipedia

UK Brimstone Anti-Armor Weapon

Credit: Duch.seb - Own work, CC BY-SA 3.0

Samsung SGR-1

Credit: AFP/Getty Images

Outline

- Motivation: Why Trustworthy and Robust AI?
- Part 1: Trustworthy Machine Learning
- Part 2: Robust Artificial Intelligence
  - Part 2A: Robust Autonomous AI Systems
  - Part 2B: Robust Human/AI Teams

Part 1: Trustworthy Machine Learning

Threats to Machine Learning Performance

- Assumption of Independent and Identically Distributed (iid) Data
- Markov Assumption in MDPs
- Closed World Assumption: The class labels are mutually-exclusive and exhaustive
- Adversarial Attacks

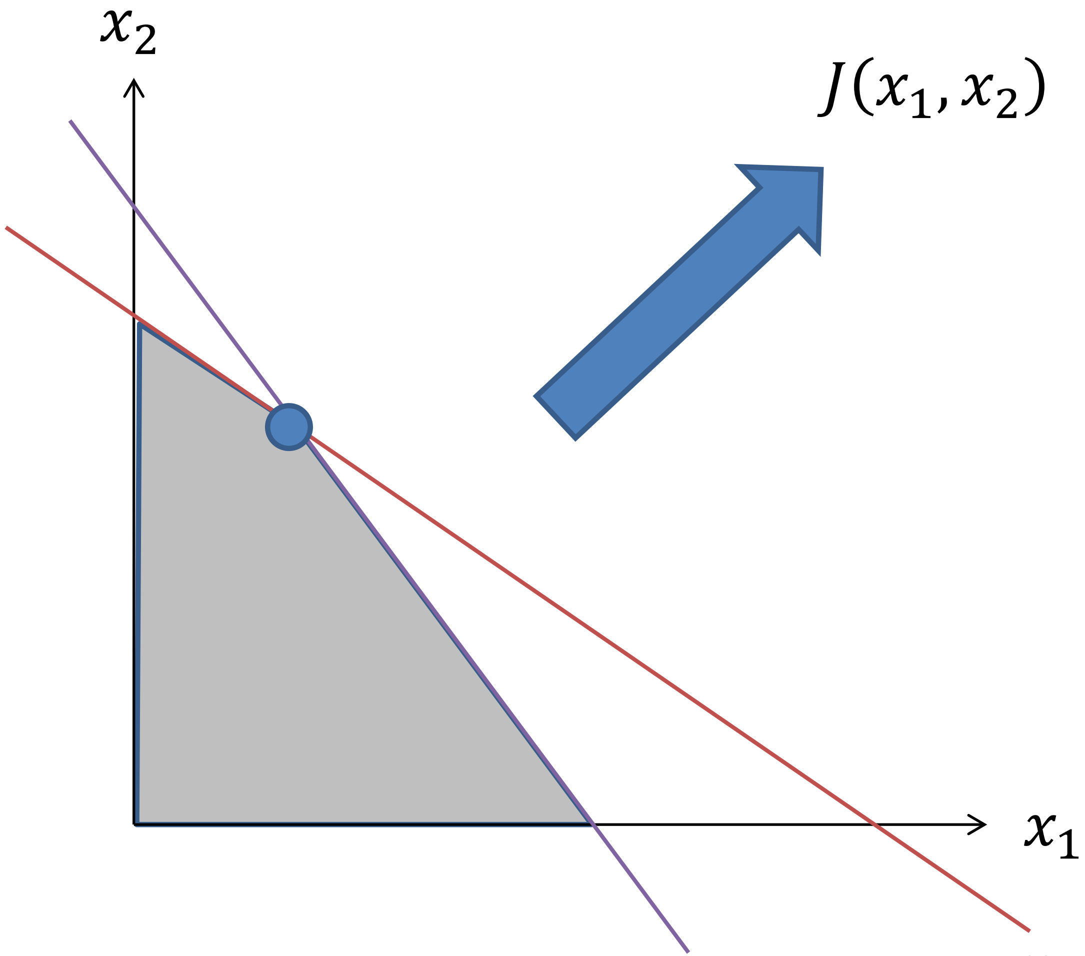
How can we be confident in the accuracy of a machine learning system?

1. Robustness by Construction
2. Self-Model of Competence
   - Calibrated Probabilities
   - Reject Option
3. Monitoring for Data Shift

Robustness by Construction: Minimaxing Against an Adversary

- Many AI reasoning problems can be formulated as optimization problems
- $\max_{x_1, x_2} \; J(x_1, x_2)$
- subject to

$$ax_1 + bx_2 \leq r$$
$$cx_1 + dx_2 \leq s$$



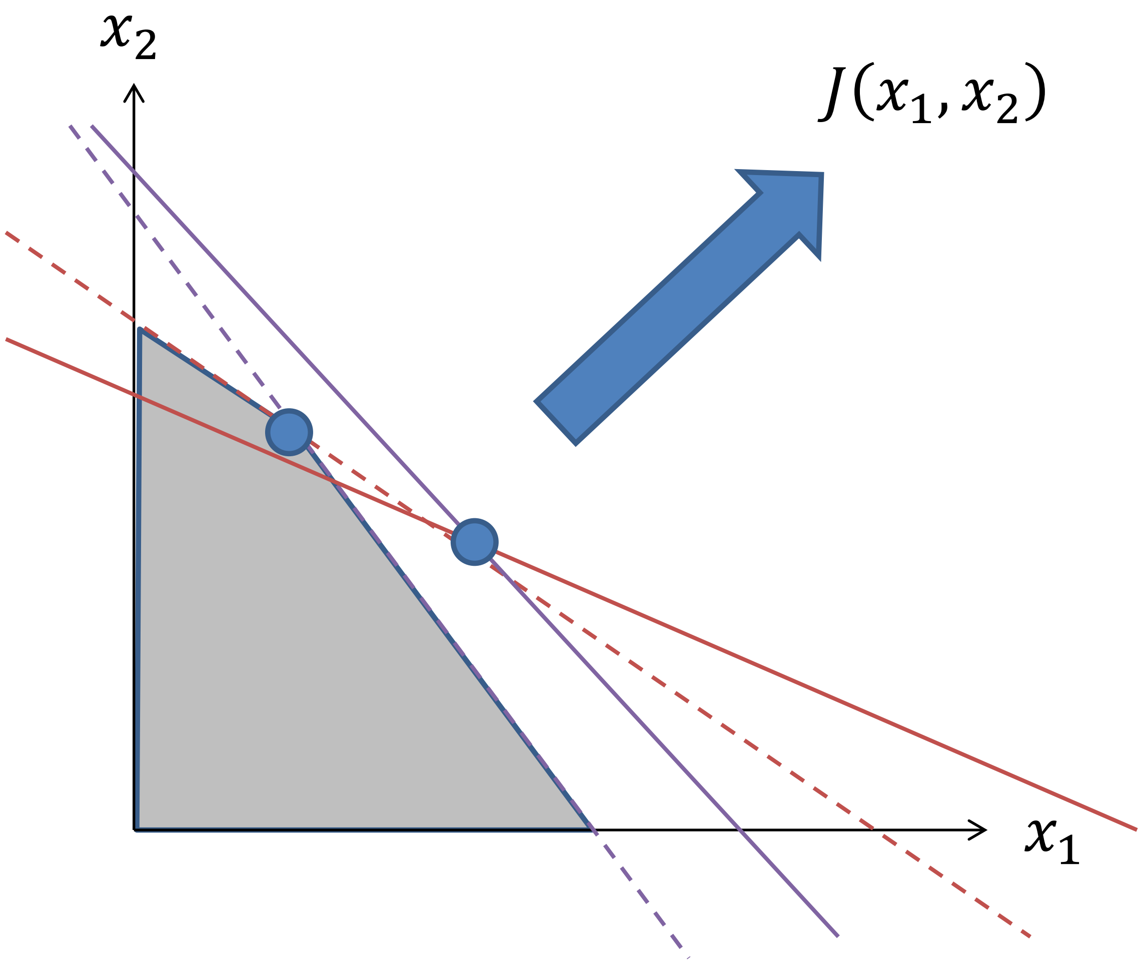
Uncertainty in the constraints

- $\max_{x_1,x_2} \; J(x_1,x_2)$
- subject to

$$ax_1 + bx_2 \le r$$
$$cx_1 + dx_2 \le s$$

- Define uncertainty regions

$$a \in U_a$$
$$b \in U_b$$
$$\dots$$
$$s \in U_s$$



Minimax against the uncertainty sets

- $\max_{x_1,x_2} \min_{a,b,c,d,r,s} J(x_1, x_2; a, b, c, d, r, s)$
- subject to

$$ax_1 + bx_2 \leq r$$
$$cx_1 + dx_2 \leq s$$
$$a \in U_a$$
$$b \in U_b$$
$$\dots$$
$$s \in U_s$$

- Problem: Solutions can be too conservative

Solution: Impose a Budget on the Adversary

- $\max_{x_1,x_2} \min_{\delta_a,\ldots,\delta_s} J(x_1, x_2; \delta_a, \ldots, \delta_s)$
- subject to

$$\begin{gathered}
(a+\delta_a)x_1 + (b+\delta_b)x_2 \leq (r+\delta_r) \\
(c+\delta_c)x_1 + (d+\delta_d)x_2 \leq (s+\delta_s) \\
\delta_a \in U_a \\
\delta_b \in U_b \\
\ldots \\
\delta_s \in U_s \\
\sum|\delta_i| \leq B
\end{gathered}$$

Bertsimas, et al. ; Ben-Tal, Nemirovski, & El Ghaoui (2009)

Robustness By Construction: Classification

- Goal: Train Deep Networks so that they are guaranteed to be robust to adversarial perturbations in the test queries
  - Measured as bound on the size of the Manhattan (ℓ_1) or Euclidean (ℓ_2) norm of the perturbation
- Approach for Linear Classifiers (e.g., linear SVM)
  - Minimax against an adversary who can perturb training point x_i by an ℓ_2 perturbation $\delta_i,\ i = 1, \dots, n$
  - This turns out to be equivalent to ℓ_2 regularization with a total budget $\lambda = \sum_i \|\delta_i\|$
  - During training, find θ to minimize $\sum_i L\big(y_i, f(x_i, \theta)\big) + \lambda\|\theta\|$
- Approach for Deep Learning: Stability Training with Noise (STN)
  - Li, Chen, Wang & Carin, 2019, arXiv 1809.03113v5
  - Find θ to minimize

$$\sum_i L\big(y_i, f(x_i, \theta)\big) + \lambda D\big(f(x_i), f(x_i + \delta_i)\big) \quad \text{where } \delta_i \sim N(0, \sigma^2 I)$$

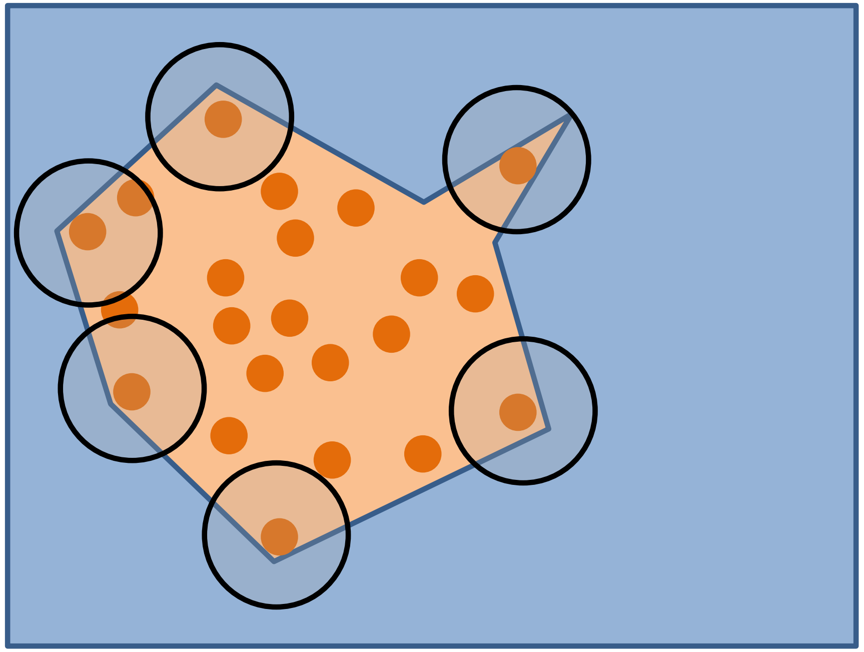
  - $D\big(f(x_i), f(x_i + \delta_i)\big) = \sum_j P(\hat{y} = j|x_i) \log P(\hat{y} = j|x_i + \delta_i)$ is the cross-entropy loss

Stability Training with Noise

- Testing: Stability testing. Measure the distribution of predictions computed with Gaussian perturbations and predict the class with highest probability

$$p_{ij} = \frac{1}{M}\sum_{m=1}^{M} \mathbb{I}\left[\hat{f}(x_i + \delta_i) = j\right] \quad \delta_i \sim N(0, \sigma^2 I)$$

$$\hat{f}_{stab}(x_i) = \arg\max_j p_{ij}$$



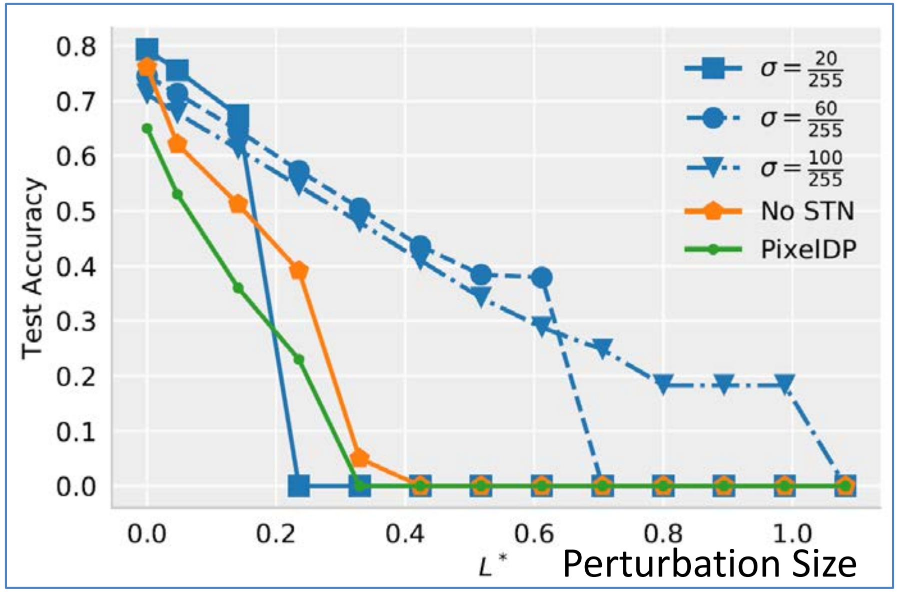
This changes the classifier to have a smoother decision boundary

Note that this is done on the test set. It provides a formal guarantee that there are no adversarial examples within a specified radius. First method that can give a non-trivial guarantee for ImageNet

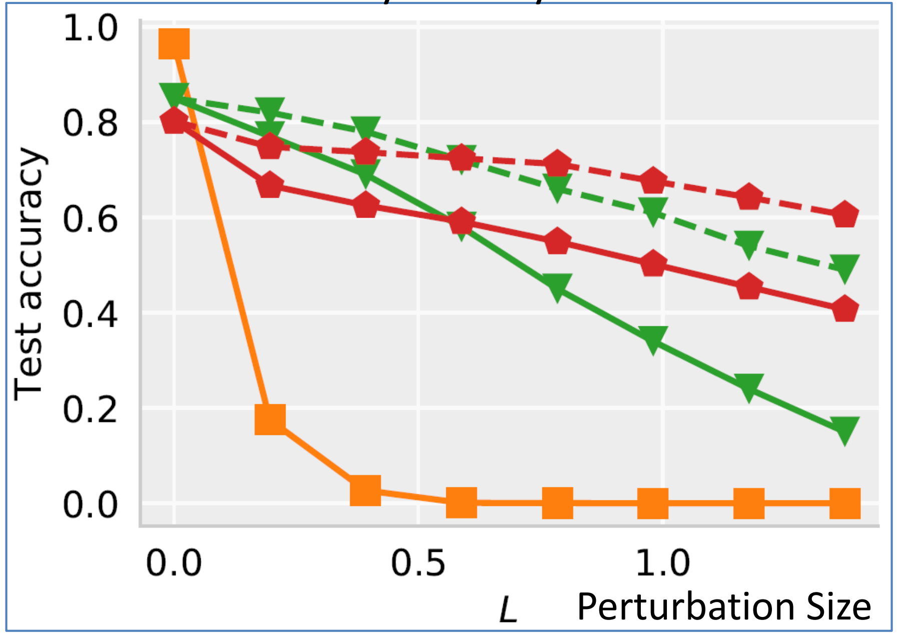
Stability Training with Noise

- Top plot: Theoretical Guarantee. Accuracy is ≥ plotted value with probability 0.95.
  - Key: **STN**, **PixelDP** (Lecuyer et al. 2018)
- Bottom plot: Experimental Accuracy. Key:
  - **STN**, **TRADES** (Zhang et al. 2019).
  - Dashes = black box (Madri attacks); solid = white box (modified Carlini & Wagner attacks)

Guaranteed Test Accuracy: CIFAR-10



Observed Accuracy: Madry Attacks



Self-Model of Competence

- Given
  - Training data S_{train}
  - Learned classifier $\hat{f}$ that outputs a probability vector $\hat{p}$
  - Test queries $S_{test} = \{x_1, \dots, x_N\}$
- Let $\hat{p}(x_i) = (\hat{p}_{i1}, \dots, \hat{p}_{ik})$ be the predicted probabilities for x_i
- Let $p(x_i) = (p_{i1}, \dots, p_{ik})$ be the true probabilities
  - includes randomness due to feature measurement error and label noise ("aleatory uncertainty")
- A classifier is well calibrated if

$$\hat{p}(x_i) = p(x_i)$$

- Weaker condition: probability of the most likely class matches

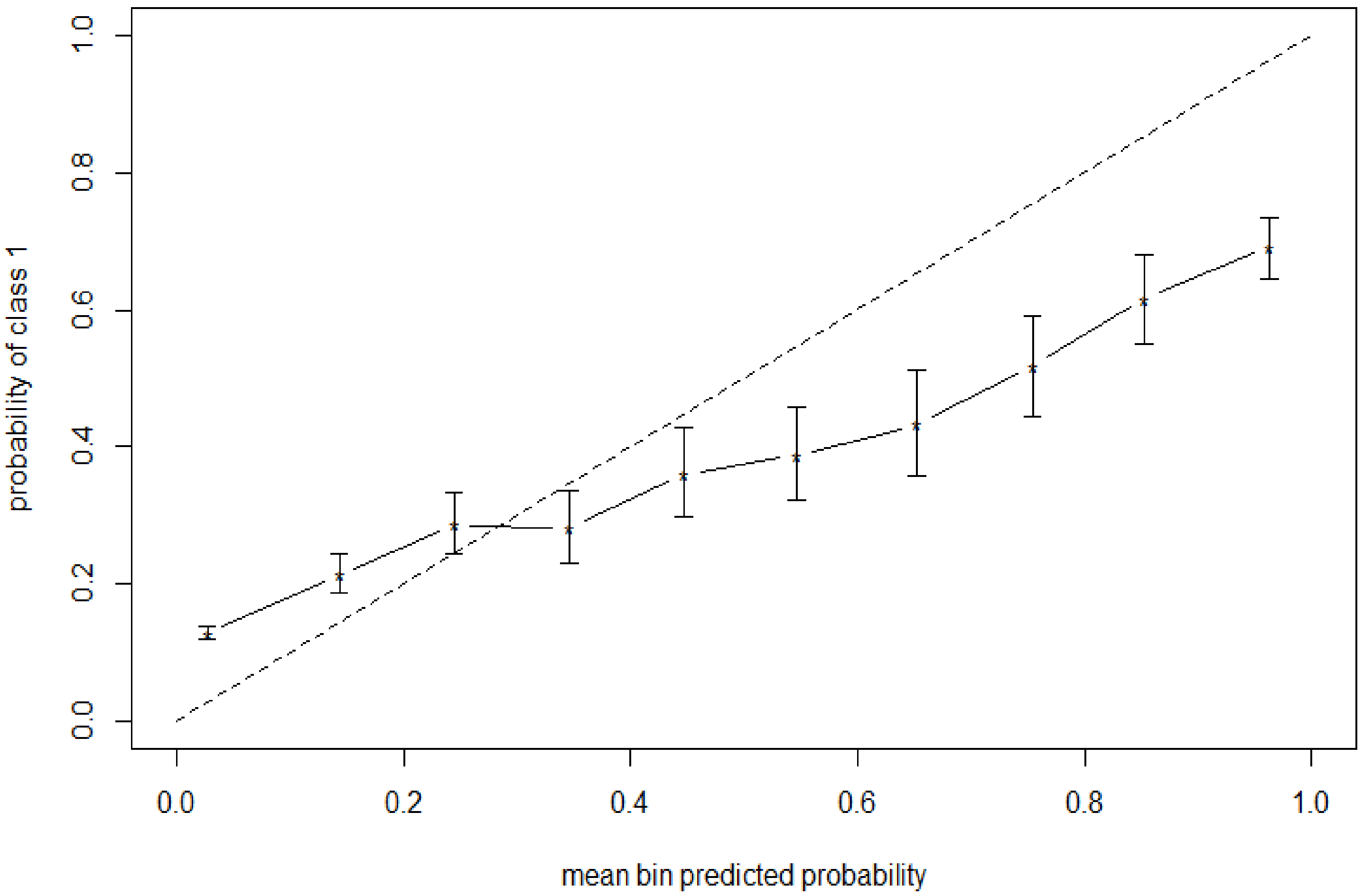
$$\arg\max_j \hat{p}_{ij} = \arg\max_j p_{ij}$$

Evaluating Calibration Using a Reliability Diagram

- On a labeled “calibration data set”, sort x_i into bins according to $\hat{p}_{i1}$
  - $[0,0.1), [0.1,0.2), \ldots$
- Compute the fraction of points in each bin for which $x_i = 1$
- Plot with confidence intervals

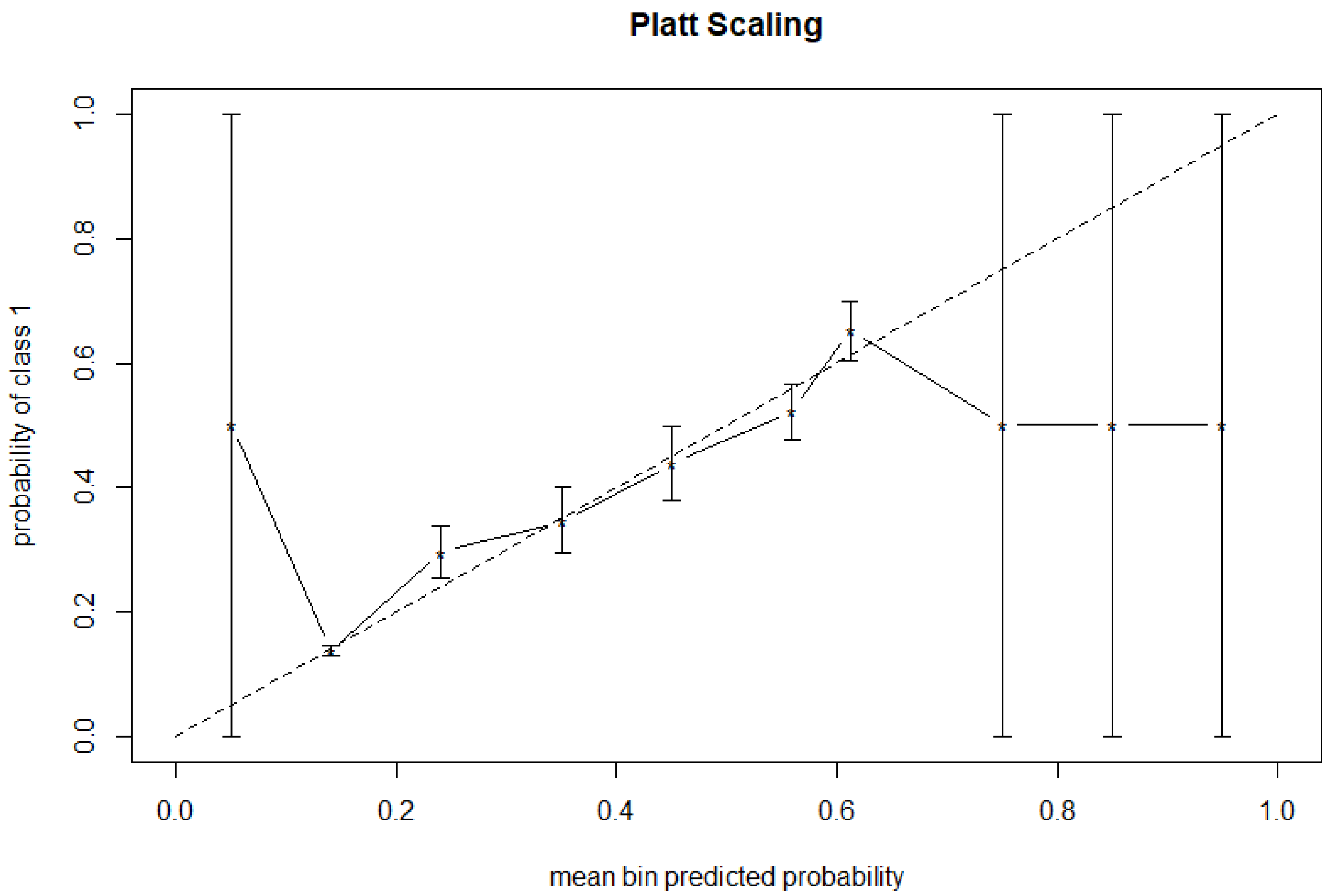
- Classifier: R gbm with 10,000 trees

UCI German Credit



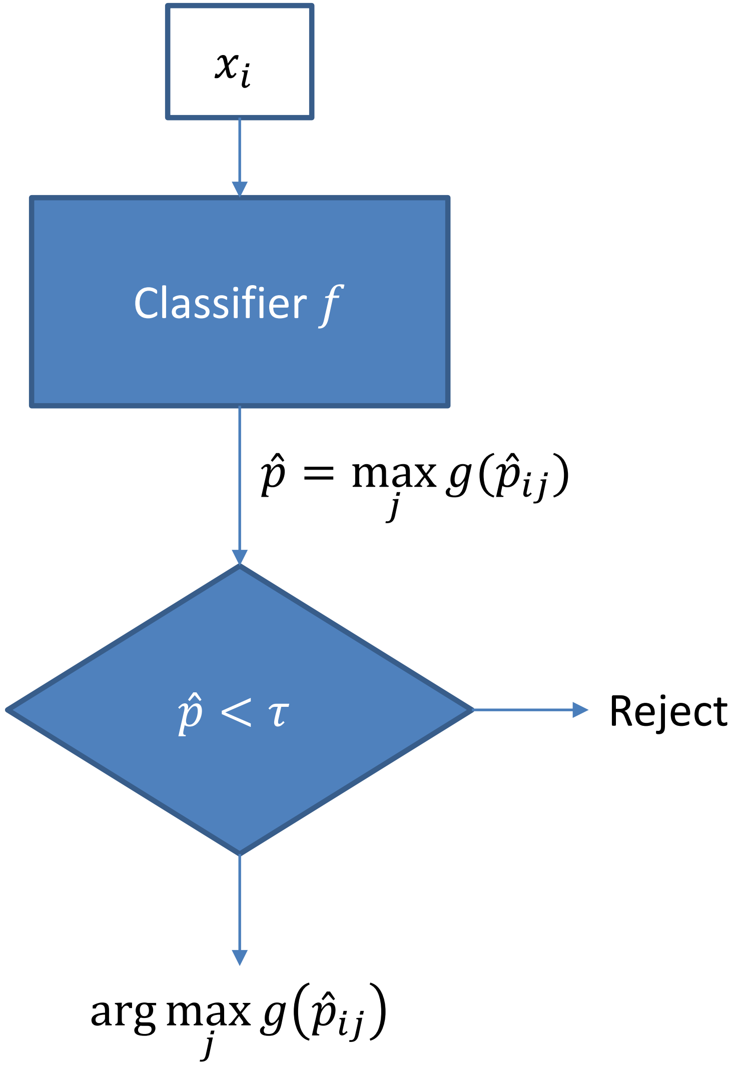
Recalibration Methods

- Several techniques have been developed for fixing poor calibration by fitting a function g such that $\tilde{p} := g(\hat{p})$ is better calibrated
  - Platt scaling (logistic regression)
  - Isotonic regression
  - Kernel logistic regression
  - Gaussian Process regression
- Reliability Diagram after Platt scaling
  - Predicted probabilities have been rescaled into the range 0.1-0.6



Using Calibrated Probabilities for Rejection

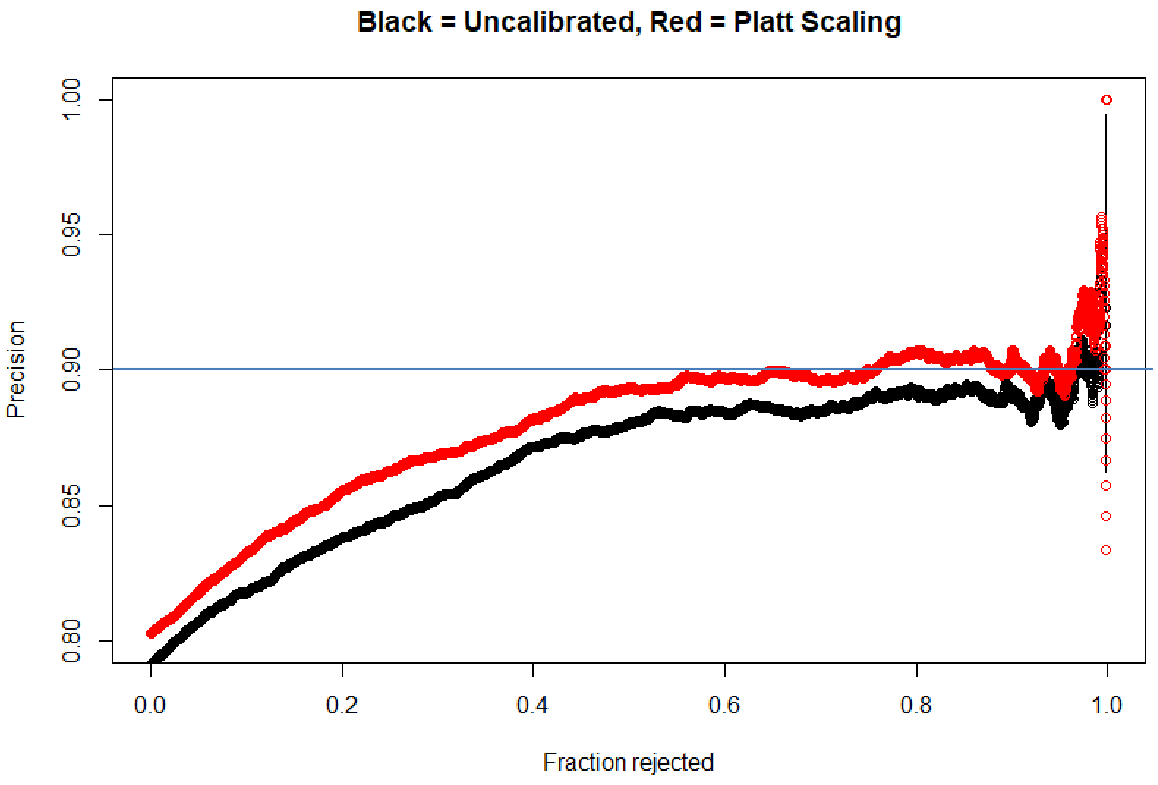
- If classifier is not confident enough, then it should reject the query

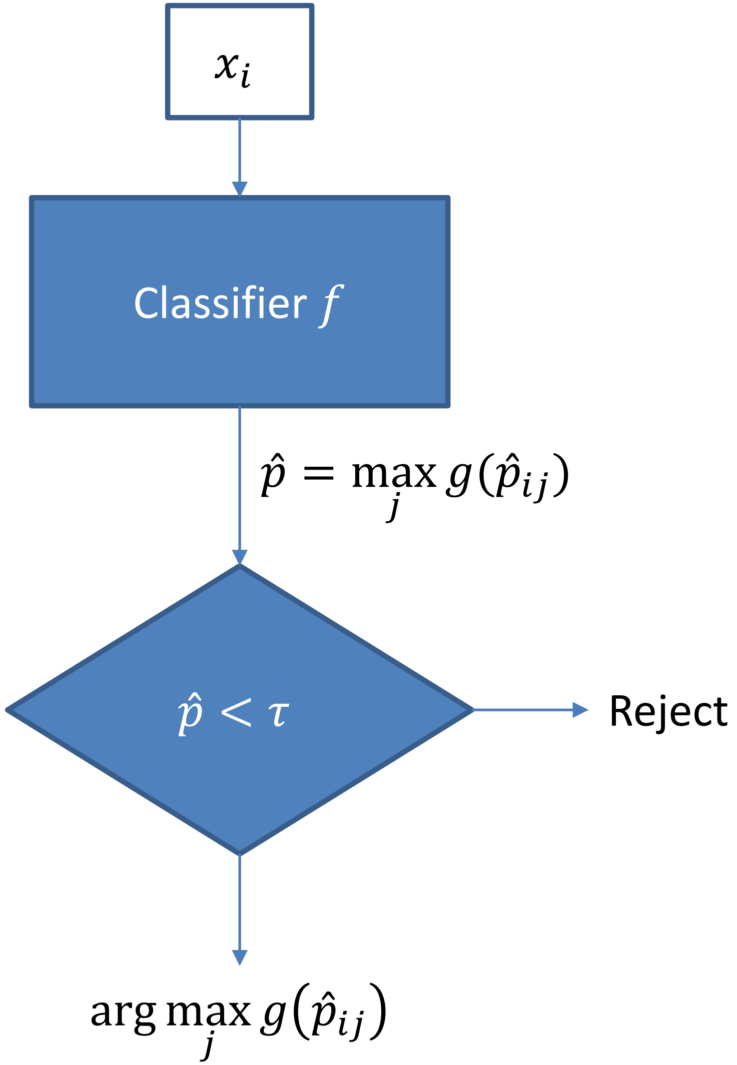


x_i
Classifier f
$\hat{p} = \max_j g(\hat{p}_{ij})$
$\hat{p} < \tau$
Reject
$\arg\max_j g\left(\hat{p}_{ij}\right)$

Using Calibrated Probabilities for Rejection

- If classifier is not confident enough, then it should reject the query
- Rejection curve. Plot precision as a function of fraction of queries rejected (by varying τ)


Black = Uncalibrated, Red = Platt Scaling


x_i

Classifier f

$\hat{p} = \max_j g(\hat{p}_{ij})$

$\hat{p} < \tau$

Reject

$\arg\max_j g\left(\hat{p}_{ij}\right)$

Monitoring for Data Shift and Novel Categories

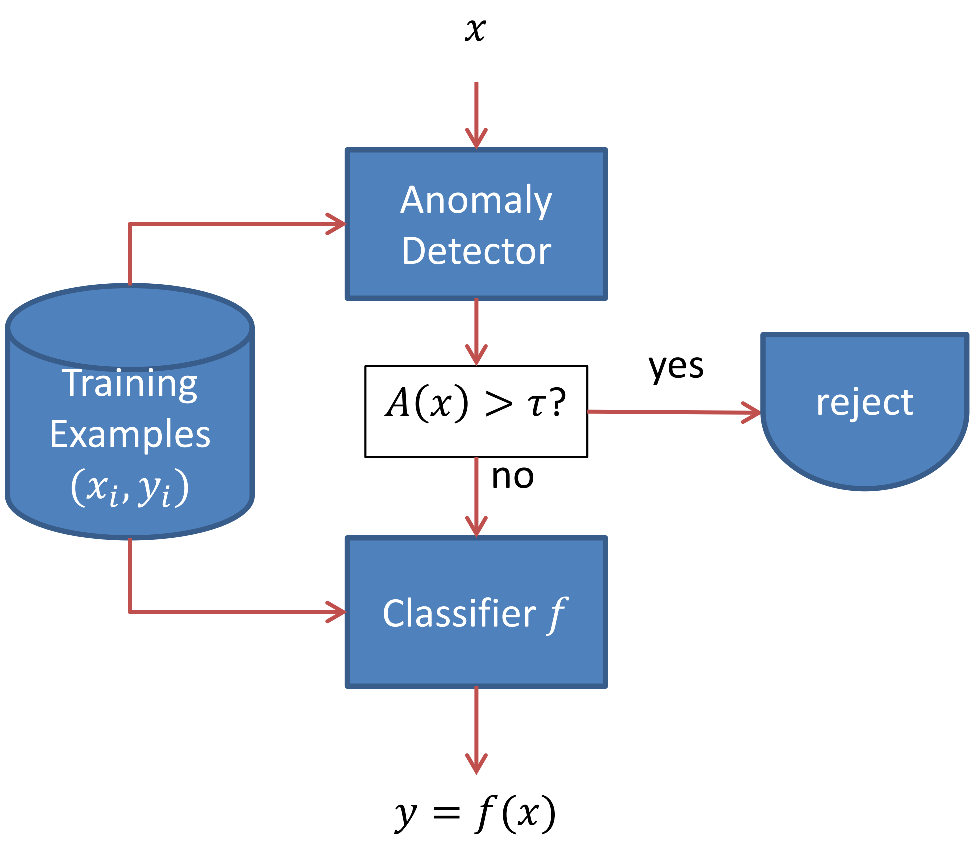
Method 1: Anomaly Detection

- Unsupervised Anomaly Detection
- Given:
  - Training data S_{train} assumed to be drawn iid from P_{train}
  - Test query x
- Decide whether x is also drawn from P_{train}

- Direct method
  - Estimate $\hat{P}_{train}$ from the training data ("density estimation")
  - Score x by "Surprise":

$$A(x) = -\log \hat{P}_{train}(x)$$

  - Difficulty: density estimation requires very large sample sizes in general



Benchmarking Practical Anomaly Detection Methods

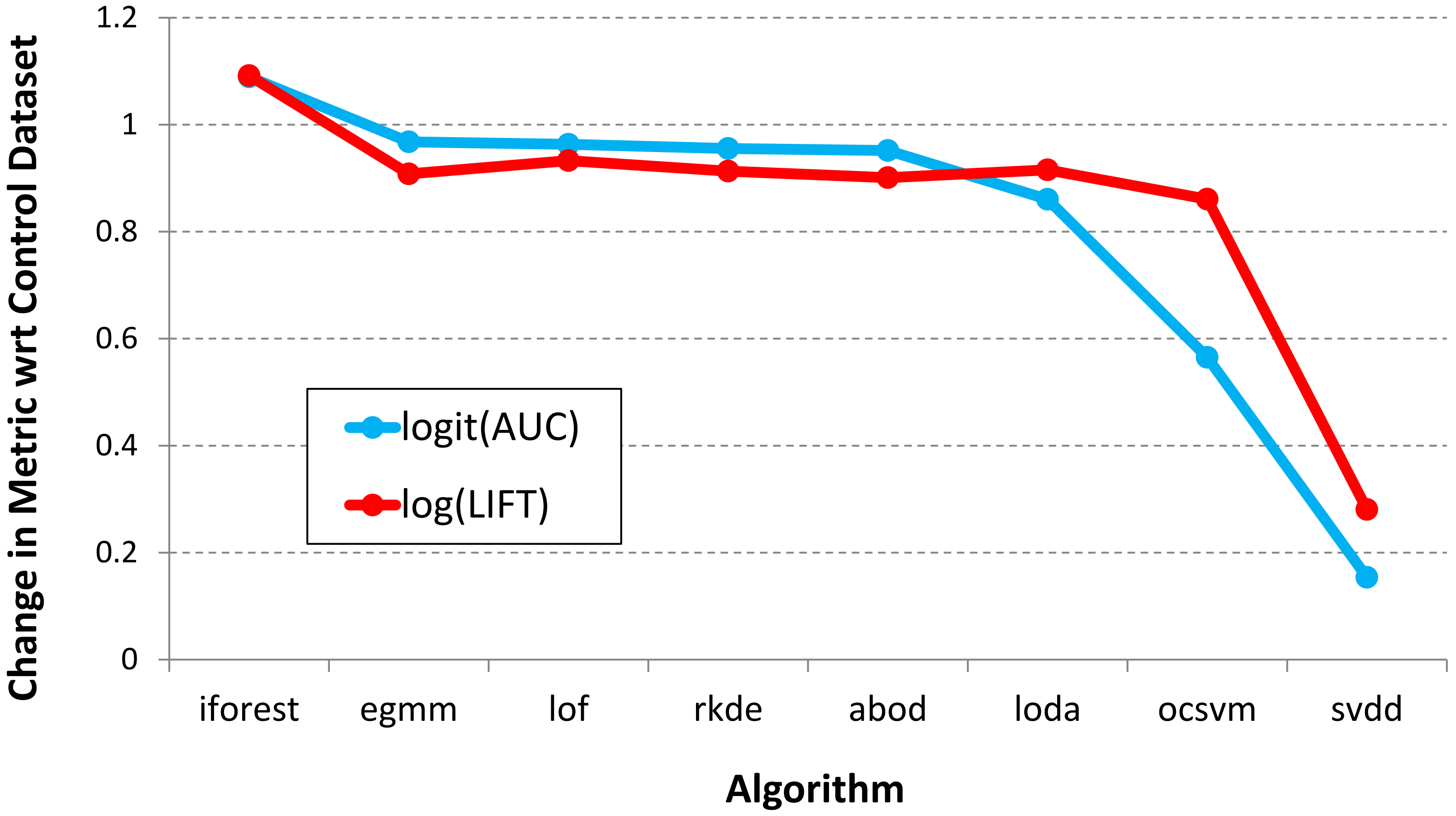
- Goal: Compare published algorithms on a robust collection of benchmarks
  - Previous comparisons suffered from small size and/or proprietary data sets

- **Density-Based Approaches**
  - RKDE: Robust Kernel Density Estimation (Kim & Scott, 2008)
  - EGMM: Ensemble Gaussian Mixture Model (our group)
- **Quantile-Based Methods**
  - OCSVM: One-class SVM (Schoelkopf, et al., 1999)
  - SVDD: Support Vector Data Description (Tax & Duin, 2004)

- **Neighbor-Based Methods**
  - LOF: Local Outlier Factor (Breunig, et al., 2000)
  - ABOD: kNN Angle-Based Outlier Detector (Kriegel, et al., 2008)
- **Projection-Based Methods**
  - IFOR: Isolation Forest (Liu, et al., 2008)
  - LODA: Lightweight Online Detector of Anomalies (Pevny, 2016)

[Emmott, Das, Dietterich, Fern, Wong, 2013; KDD ODD-2013]
[Emmott, Das, Dietterich, Fern, Wong. 2016; arXiv 1503.01158v2]

Anomaly Detection Benchmark Results



iForest was best; quantile methods were worst; all others approximately equal

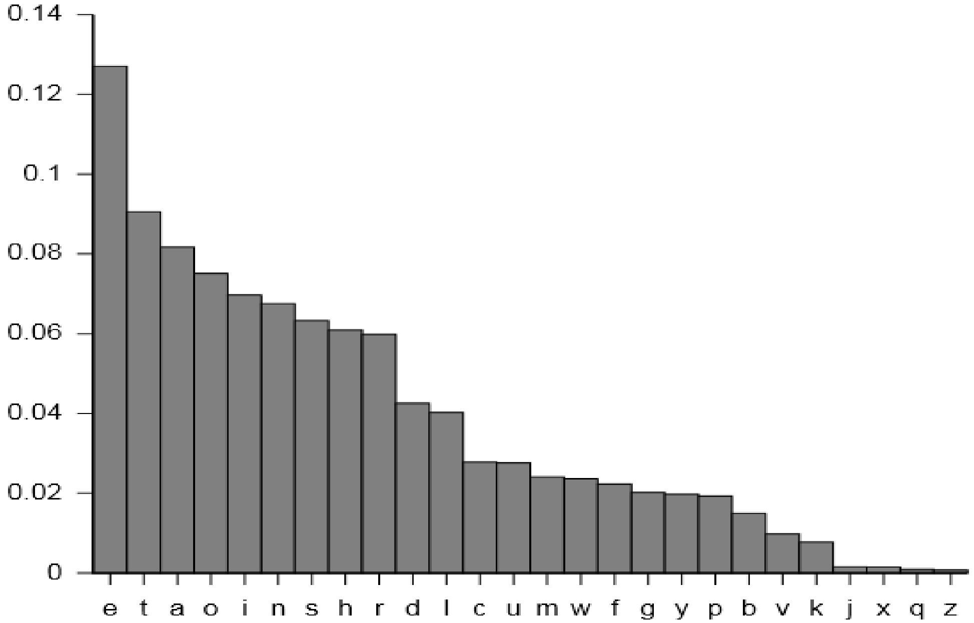
Anomaly Detection Challenges

- High-dimensional spaces are inherently difficult
  - Can we assume the data live in a lower-dimensional subspace?
- Image and video data
  - Need to discover the lower-dimensional space
- Promising directions
  - Auto-encoders and generative models (VAE, RAE, BiGAN)
  - Neural Rendering Model
  - Extending existing methods to work with time series

Data Shift Detection:
Method 2: Monitor the Distribution of Predicted Classes

- Supervised classification
  - On training data, measure expected class frequencies
  - Detect departures from these on test data
- Mismatch can indicate a change in the class distribution or a failure in the classifier

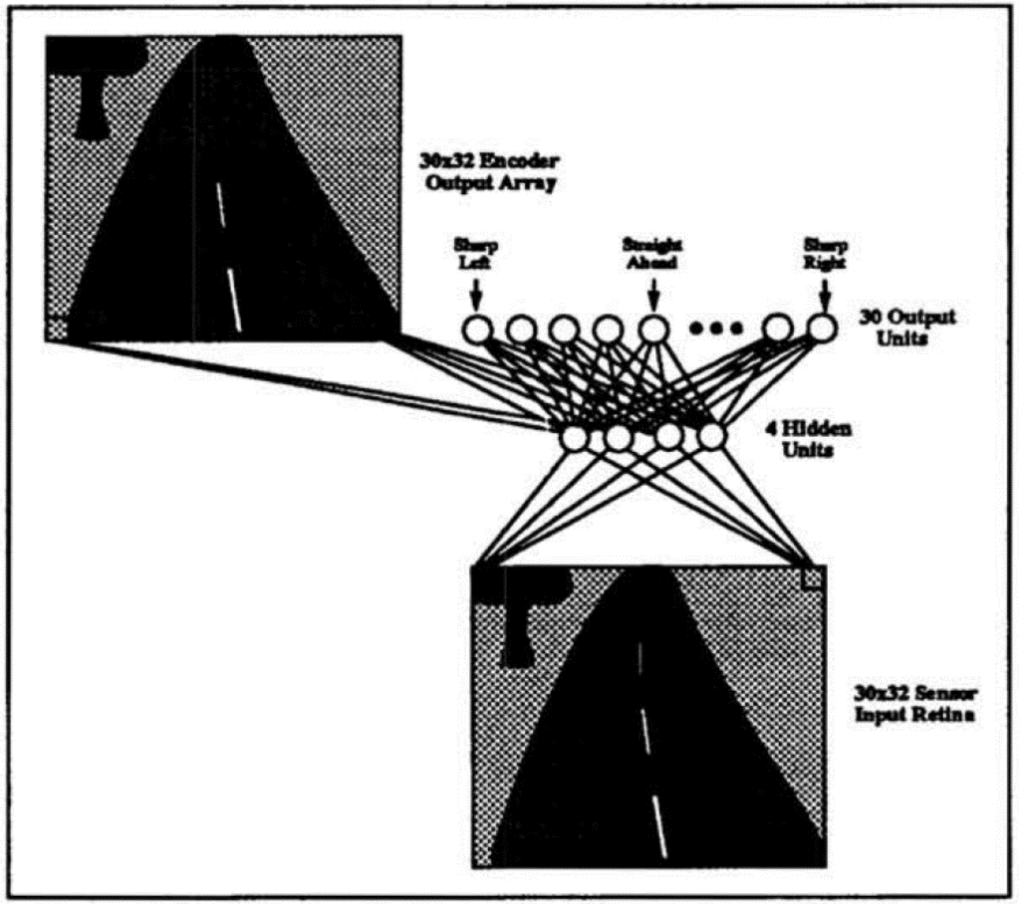
Letter frequencies in English



Credit: Nandhp, Wikipedia

Method 3: Monitor Auxiliary Tasks

- ALVINN auto-steer system
- Main task: Determine steering command
- Auxiliary task: Predict input image
- Perform both tasks with the same hidden layer information



Pomerleau, NIPS 1992

Data Shift Detection and Repair
Method 4: Two-Sample Testing

- Compare S_{train} to S_{test}
- Technique 1: Two-sample Tests
  - Kernel Two-sample Test based on Maximum Mean Discrepancy (Gretton, et al.)
- Technique 2: Train a classifier
  - Label S_{train} as class 0
  - Label S_{test} as class 1
  - Train a classifier and evaluate via cross-validation
  - If the classifier can do better than random guessing, then we have data shift

Outline

- Motivation: Why Trustworthy and Robust AI?
- Part 1: Trustworthy Machine Learning
- Part 2: Robust Artificial Intelligence
  - Part 2A: Robust Autonomous AI Systems
  - Part 2B: Robust Human/AI Teams

Goal: Robust Artificial Intelligence

- Definition: System remains safe and successful in spite of
  - Errors in the problem formulation
  - Errors in authored or learned models
  - Sensor failures
  - Changes in the system and in the world
  - Errors by human operators
  - Breakdowns in human teams
  - Cyberattack

High Reliability Organizations

Todd LaPorte, Gene Rochlin, and Karlene Roberts

- Studied several high reliability human teams
  - Air Traffic Control
  - Nuclear power plant operations
  - Aircraft Carrier flight deck operations

- Claim: Accidents can be prevented through organizational design, culture, management, and human choices

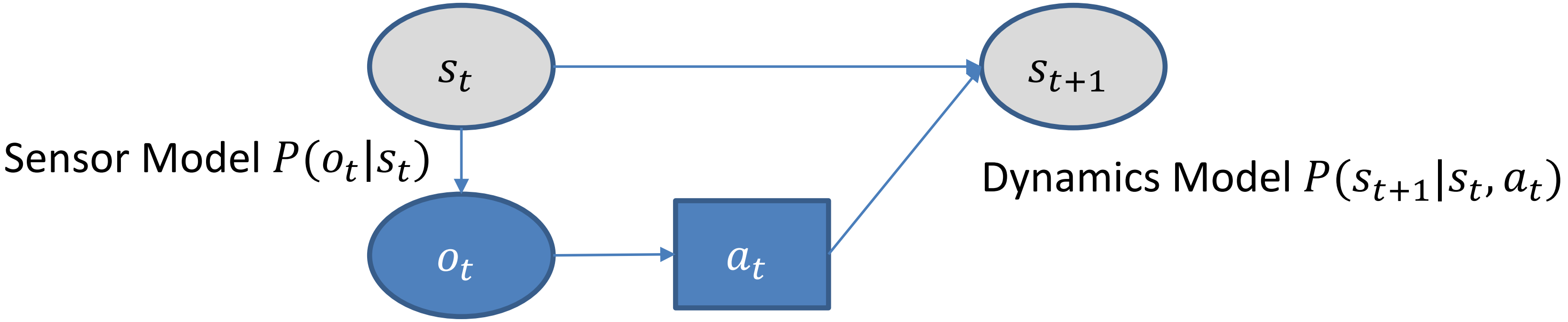
- Impact:
  - Patient safety movement
  - Cockpit resource management

Properties of High Reliability Organizations

- Preoccupation with failure
  - Fundamental belief that the system has unobserved failure modes
  - Treat anomalies and near misses as symptoms of a problem with the system
- Reluctance to simplify interpretations
  - Comprehensively understand the situation
- Sensitivity to operations
  - Maintain continuous situational awareness
- Commitment to resilience
  - Develop the capability to detect, contain, and recover from errors. Practice improvisational problem solving
- Deference to expertise
  - During a crisis, authority migrates to the person who can solve the problem, regardless of their rank

PART 2A: AUTONOMOUS AI SYSTEMS

Maintain Situational Awareness


s_t
s_{t+1}
Sensor Model $P(o_t|s_t)$
Dynamics Model $P(s_{t+1}|s_t, a_t)$
o_t
a_t

- Maintain a probability distribution $P(s_t)$ over the state of the system
- Collect the observations o_t
- Compute updated distribution:
$$P(s_t|o_t) \propto P(o_t|s_t)P(s_t)$$
- Choose the action a_t
- Predict next state distribution:

$$P(s_{t+1}|o_t, a_t) = \sum_{s_t} P(s_{t+1}|a_t, s_t)P(s_t|o_t)$$

- Methods:
  - Kalman filter
  - Particle filters
  - Expectation propagation
  - Variational approximations
  - etc.

Detect Anomalies and Near Misses

Detecting Anomalies

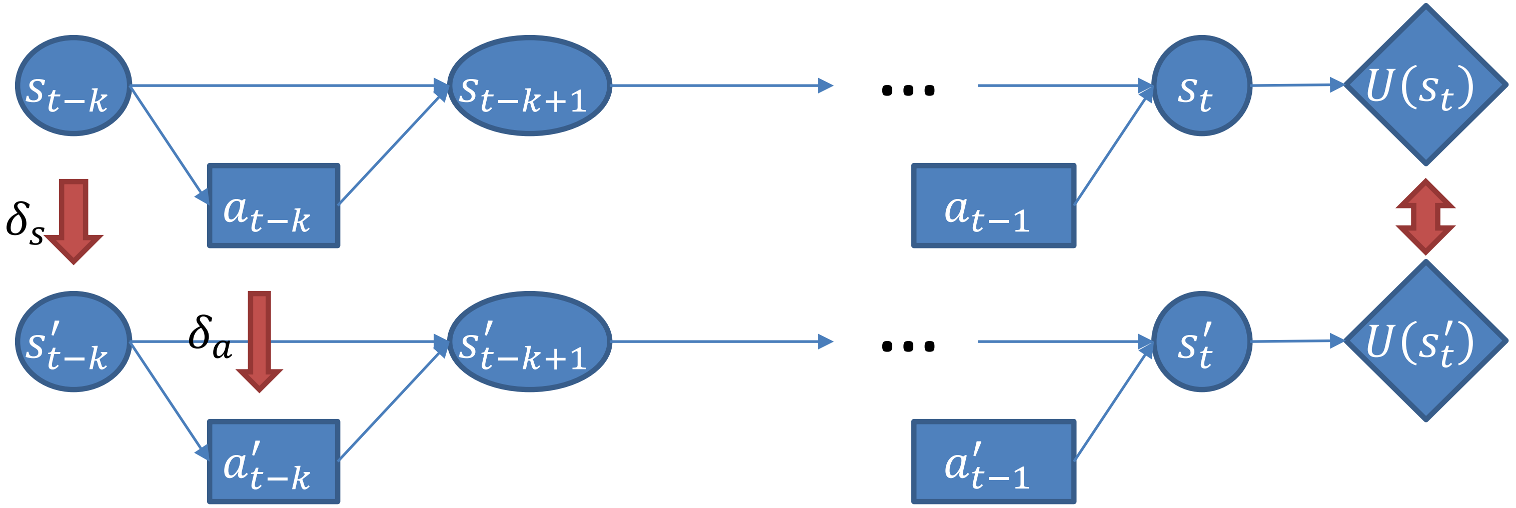
- Compute the “surprise” of the observed o_{t+1}
- Predicted distribution of o_{t+1}:

$$P(o_{t+1}|o_t, a_t) = \sum_{s_{t+1}} P(s_{t+1}|o_t, a_t)P(o_{t+1}|s_{t+1})$$

- Anomaly Score:

$$-\log P(o_{t+1}|o_t, a_t)$$

Detecting Near Misses



- Suppose we have a utility function $U(s)$ over states
- Counterfactual Notion: Perturb s_{t-k} and/or a_{t-k}
- Near Miss:

$$U(s'_t) \ll U(s_t)$$

- Detecting near misses is under-studied; requires causal model
- Should anticipate them and act to prevent them (ACAS-X)

Explaining Anomalies and Near Misses: Research Challenges

- Open-ended space of hypotheses
  - Effects of exogenous variables / unknown external agents?
    - what external agents might exist and why would they be affecting our system?
  - Sensor failures and/or inadequate sensors
    - why didn’t we detect the anomaly or near miss earlier?
  - Model failures (dynamics and sensor models)
    - did the system structure change? (broken pipe? stuck valve?)
- Promising work
  - Model-based diagnosis including performing information-gathering actions

Improvising Solutions: Finding Repairs and Workarounds

- Approaches
  - Update dynamics and sensor models and then apply planning algorithms?
  - Mark aspects of the models as unreliable and seek a plan that does not depend on those aspects?
  - Always plan conservatively to be robust to model errors?

Summary: Autonomous AI

	Assessment
Situational Awareness	A mature methods
Detect Anomalies and Near Misses	B high-dimension, dynamics
Explain Anomalies and Near Misses	D only basic techniques
Improvise Solutions	F

PART 2B: AI + HUMAN TEAMS

AI and Human Teams

- Even very powerful AI systems will be surrounded by a human team that will determine
  - What goals to give it
  - What degree of autonomy to permit it
  - When to trust it
  - What degree of learning/adaptation to allow

- How can the combined AI + Human Team be safe and robust?
  - Reconsider each aspect of high-reliability organizations from an interactive perspective

Situational Awareness: Past Failures

- Autopilot Tunnel Vision: Aircraft autopilot not aware of air traffic control instructions
  - Co-pilot must continually update the autopilot’s waypoints based on ATC interactions
  - This load increases in high-traffic/high-risk situations
  - Co-pilot loses awareness of other aspects of the system
- Autopilot Fails to Communicate Situation
  - Colgan Air 3407 crash near Buffalo
  - Autopilot was compensating for aircraft icing, but pilots were not aware of this
  - Eventually autopilot was forced to hand control back to pilots
  - Their lack of situational awareness led to crash (“decompensation failure”)
- Autopilot Over-Communicates
  - Hundreds of unimportant alarms
  - Complex displays that bury important information
- Humans Misunderstand Internal State of Autonomous System
  - USS John McCain collision: team thought single slider was controlling both engines, but it was controlling only one
  - Caused ship to turn into the course of an oncoming ship

Requirements for Robust Situational Awareness

- AI system should have sufficient sensing
  - state of world including other agents
  - state of the system being controlled
  - state of its human team
- Human team and AI system should establish and maintain a shared mental model
  - AI system should reason about what the users know and do not know and communicate strategically
  - Humans need a good mental model of the AI system's beliefs about the situation
  - AI system needs to be able to explain its beliefs to humans
  - Careful design of user interface is critical

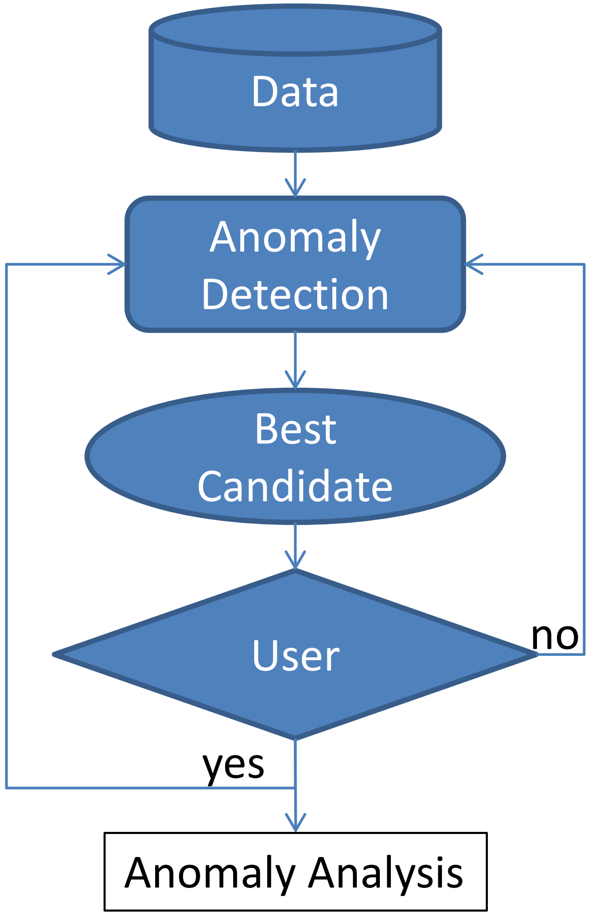
Anomaly and Near Miss Detection

- Existing methods are highly local
  - sensor readings out of standard range
  - violations of minimum separation (air-to-air, air-to-ground, car-to-car)
- Need more and better anticipation of problems
  - model the behavior of other agents (including team members)
  - project system state many steps into the future and evaluate
- Incorporate interactive anomaly detection

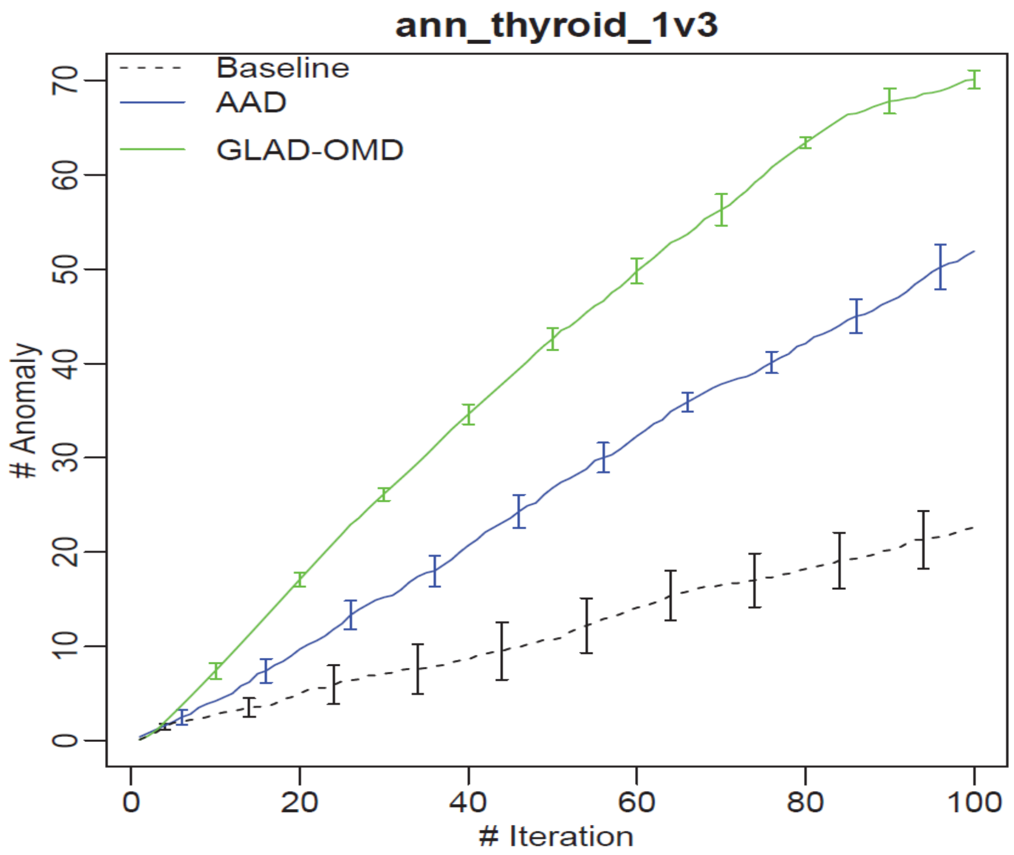
Incorporating User Feedback: Initial Work

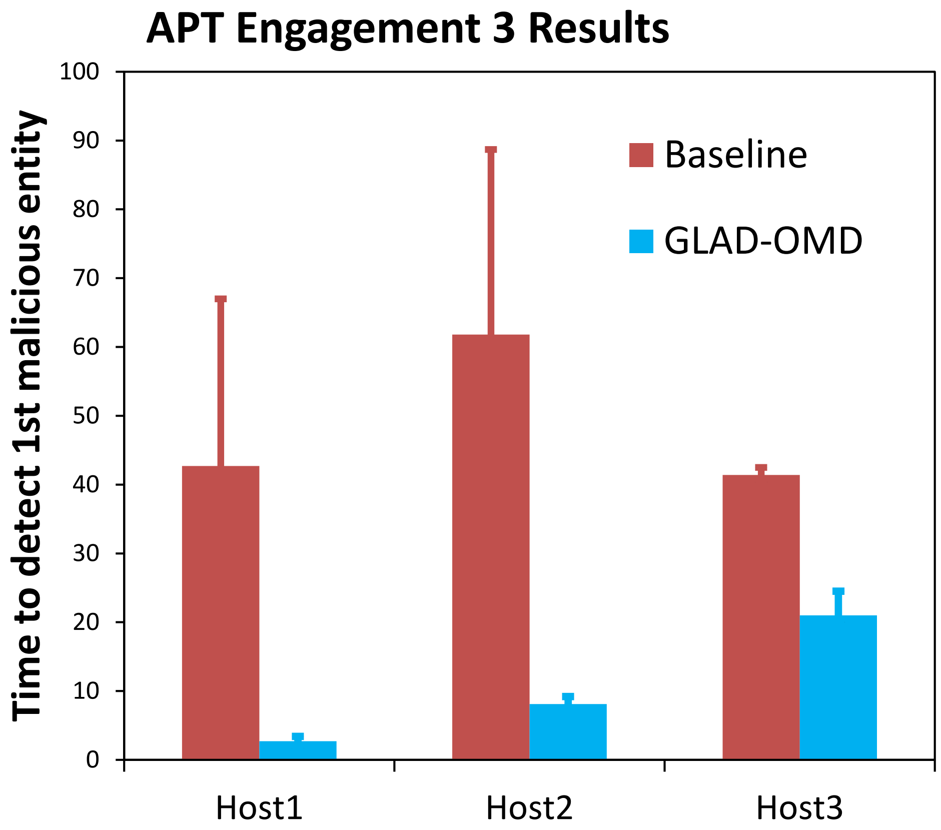
- Show top-ranked candidate to the user
- User labels candidate
- Label is used to update the anomaly detector

[Das, et al, ICDM 2016]
GLAD-OMD [Siddiqui, et al., KDD 2018]



User Feedback Yields Big Improvements in Anomaly Discovery





Explaining Anomalies and Near Misses

- Existing anomaly explanations are purely statistical
  - “This credit card transaction is anomalous because it was very large compared to this customer’s normal behavior”
- Root cause analysis
  - “Customer just purchased a house and is buying furniture for it”
  - Must consider a broader set of hypotheses than in normal state updating
  - May lack dynamics and observation models for this broader space

Improvisational Problem Solving

- Human users and AI system collaborate to find solutions
- Humans “think outside the box” to enlarge the problem space
- How can the AI system help humans reason about this larger problem space?
  - Verify that proposed plan does not violate any known system limits or lead to bad system states within the AI’s narrow problem space?
  - Can humans communicate the larger space to the AI system so that it can reason about it?
  - Explain to humans how the AI system would behave if permitted autonomy
- Existing work:
  - Mixed-initiative Planning

Mixed-Initiative Planning

- Agenda of activities that need to be planned
- User-invoked planning operators
  - Plan all: fully automated planning
  - Plan selected goals: incrementally add one or more activities to the emerging plan
  - Expand selected subgoal
  - Create a plan sketch (commit to some activities, possibly at different levels of abstraction)
- User plan editing
  - Move an activity to a different time while disturbing existing steps as little as possible
  - Add/delete activity
  - Delete or relax a constraint
  - Tentatively fix a decision but note that if additional information arrives (e.g., weather forecast) then this decision should be revisited
- System continually checks that all constraints are satisfied and makes changes to satisfy resource constraints and mutual exclusion constraints

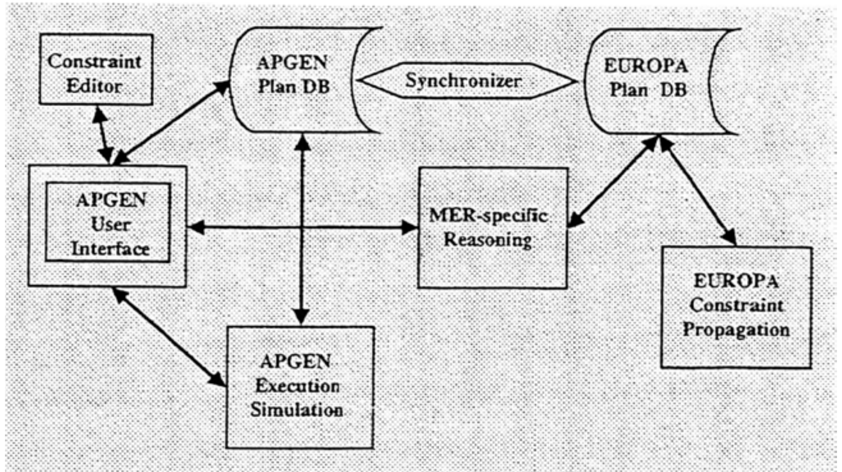
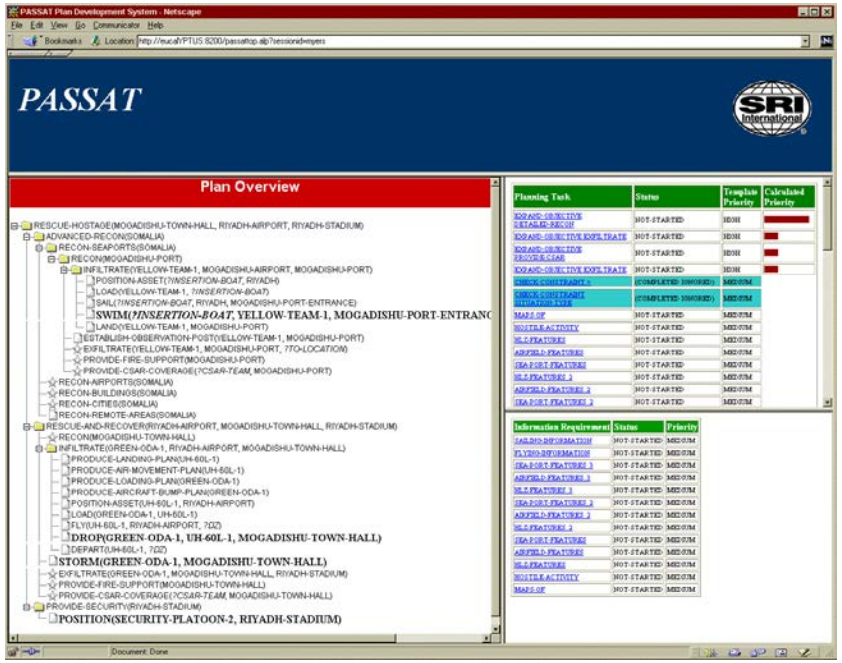


Figure 2: MAPGEN Architecture

MAPGEN: Bresina et al., 2005


PASSAT Plan Development System - Netscape
File Edit View Go Communicator Help
Bookmarks Location
PASSAT
SRI International
Document: Done

Plan Overview

```
RESCUE-HOSTAGE(MOGADISHU-TOWN-HALL, RIYADH-AIRPORT, RIYADH-STADIUM)
  ADVANCED-RECON(SOMALIA)
    RECON-SEAPORTS(SOMALIA)
      RECON(MOGADISHU-PORT)
        INFILTRATE(YELLOW-TEAM-1, MOGADISHU-AIRPORT, MOGADISHU-PORT)
          POSITION-ASSET(?INSERTION-BOAT, RIYADH)
          LOAD(YELLOW-TEAM-1, ?INSERTION-BOAT)
          SAIL(?INSERTION-BOAT, RIYADH, MOGADISHU-PORT-ENTRANCE)
          SWIM(?INSERTION-BOAT, YELLOW-TEAM-1, MOGADISHU-PORT-ENTRANC
          LAND(YELLOW-TEAM-1, MOGADISHU-PORT)
        ESTABLISH-OBSERVATION-POST(YELLOW-TEAM-1, MOGADISHU-PORT)
        EXFILTRATE(YELLOW-TEAM-1, MOGADISHU-PORT, ?TO-LOCATION)
        PROVIDE-FIRE-SUPPORT(MOGADISHU-PORT)
        PROVIDE-CSAR-COVERAGE(?CSAR-TEAM, MOGADISHU-PORT)
    RECON-AIRPORTS(SOMALIA)
    RECON-BUILDINGS(SOMALIA)
    RECON-CITIES(SOMALIA)
    RECON-REMOTE-AREAS(SOMALIA)
  RESCUE-AND-RECOVER(RIYADH-AIRPORT, MOGADISHU-TOWN-HALL, RIYADH-STADIUM)
    RECON(MOGADISHU-TOWN-HALL)
    INFILTRATE(GREEN-ODA-1, RIYADH-AIRPORT, MOGADISHU-TOWN-HALL)
      PRODUCE-LANDING-PLAN(UH-60L-1)
      PRODUCE-AIR-MOVEMENT-PLAN(UH-60L-1)
      PRODUCE-LOADING-PLAN(GREEN-ODA-1)
      PRODUCE-AIRCRAFT-BUMP-PLAN(GREEN-ODA-1)
      POSITION-ASSET(UH-60L-1, RIYADH-AIRPORT)
      LOAD(GREEN-ODA-1, UH-60L-1)
      FLY(UH-60L-1, RIYADH-AIRPORT, ?DZ)
      DROP(GREEN-ODA-1, UH-60L-1, MOGADISHU-TOWN-HALL)
      DEPART(UH-60L-1, ?DZ)
    STORM(GREEN-ODA-1, MOGADISHU-TOWN-HALL)
    EXFILTRATE(GREEN-ODA-1, MOGADISHU-TOWN-HALL, RIYADH-STADIUM)
    PROVIDE-FIRE-SUPPORT(MOGADISHU-TOWN-HALL)
    PROVIDE-CSAR-COVERAGE(?CSAR-TEAM, MOGADISHU-TOWN-HALL)
  PROVIDE-SECURITY(RIYADH-STADIUM)
    POSITION(SECURITY-PLATOON-2, RIYADH-STADIUM)
```

Planning Task	Status	Template Priority	Calculated Priority
EXPAND-OBJECTIVE DETAILED-RECON	NOT-STARTED	HIGH	▬
EXPAND-OBJECTIVE EXFILTRATE	NOT-STARTED	HIGH	▬
EXPAND-OBJECTIVE PROVIDE-CSAR	NOT-STARTED	HIGH	▬
EXPAND-OBJECTIVE EXFILTRATE	NOT-STARTED	HIGH	▬
CHECK-CONSTRAINT =	(COMPLETED IGNORED)	MEDIUM	
CHECK-CONSTRAINT SITUATION-TYPE	(COMPLETED IGNORED)	MEDIUM	
MAPS-OF	NOT-STARTED	MEDIUM	
HOSTILE-ACTIVITY	NOT-STARTED	MEDIUM	
HLZ-FEATURES	NOT-STARTED	MEDIUM	
AIRFIELD-FEATURES	NOT-STARTED	MEDIUM	
SEA-PORT-FEATURES	NOT-STARTED	MEDIUM	
HLZ-FEATURES 2	NOT-STARTED	MEDIUM	
AIRFIELD-FEATURES 2	NOT-STARTED	MEDIUM	
SEA-PORT-FEATURES 2	NOT-STARTED	MEDIUM	

Information Requirement	Status	Priority
SAILING-INFORMATION	NOT-STARTED	MEDIUM
FLYING-INFORMATION	NOT-STARTED	MEDIUM
SEA-PORT-FEATURES 3	NOT-STARTED	MEDIUM
AIRFIELD-FEATURES 3	NOT-STARTED	MEDIUM
HLZ-FEATURES 3	NOT-STARTED	MEDIUM
SEA-PORT-FEATURES 2	NOT-STARTED	MEDIUM
AIRFIELD-FEATURES 2	NOT-STARTED	MEDIUM
HLZ-FEATURES 2	NOT-STARTED	MEDIUM
SEA-PORT-FEATURES	NOT-STARTED	MEDIUM
AIRFIELD-FEATURES	NOT-STARTED	MEDIUM
HLZ-FEATURES	NOT-STARTED	MEDIUM
HOSTILE-ACTIVITY	NOT-STARTED	MEDIUM
MAPS-OF	NOT-STARTED	MEDIUM

PASSAT: Myers, et al., 2002

Decision Making

- Person with the relevant expertise should make the final decision
  - Course of action
  - Decision to delegate actions to AI system

Past Failures

- AI capabilities and limitations are unclear to humans
  - Humans trust AI autonomy when they should not
    - Gulf War Patriot Missile Fratricide
      - New crew operating unfamiliar equipment
      - Broken radio communication with other teams
      - Patriot missile system incorrectly interpreted returning British fighter jet as incoming ballistic missile
      - Crew trusted the system, launched defensive missile: 2 killed
    - Iran-Iraq War AEGIS autonomous ship defense system
      - AEGIS and crew misinterpreted civilian aircraft as incoming attacker despite IFF transponder signal
      - Armed AEGIS which then shot down the aircraft: 290 killed

- AI current and future behavior is difficult for humans to predict
  - Symptom: Humans continually monitor AI system behavior and prepare to intervene at any moment

Past Failures (2)

- Teamwork failures lead to accidents
  - USS Fitzgerald collision with ACX Crystal
    - Poor communications including failure to use advanced navigation aides led to loss of situational awareness
    - Collision killed 7

Requirements for Human + AI Teams

- AI System needs to monitor functioning of human team
  - Detect communication failures
  - Detect misunderstandings (failures of shared mental model)
- AI System needs to know when to defer to human expertise
  - Model the expertise of each team member
  - Know whom to engage to obtain information or make a decision
- If human teamwork is breaking down, AI system should abort mission and switch to a safe backup plan

Summary: Human + AI Teams

	Assessment
Situational Awareness	C poor UI, poor communication
Detect Anomalies and Near Misses	C user feedback to anomaly detection
Explain Anomalies and Near Misses	D only basic techniques
Improvise Solutions	D mixed-initiative planning

SUMMARY

Part 1: Trustable Machine Learning

- Robustness by Construction
  - Budgeted Robust Optimization provides a useful framework
  - Regularization via Stability Training with Noise (STN) provides guarantees and a practical algorithm
  - Optimizing for CVaR provides robustness in MDP planning and reinforcement learning
- Self-Model of Competence
  - Calibrated Probabilities can be obtained by post-processing classifier scores
  - Classifiers can decide when to reject a query by thresholding calibrated probabilities
- Monitoring for Data Shift
  - Many methods for detecting data shift: anomaly detection, shift in distribution of predicted classes, shift in auxiliary tasks, two-sample test, classifier discrimination test

Part 2: Robust AI Systems

- High Reliability Human Organizations provide a model for achieving robustness in complex, safety-critical tasks

- Part 2A: Autonomous AI Systems
  - Maintain Situational Awareness (well understood)
  - Detecting Anomalies and Near Misses (many open questions)
  - Explaining Anomalies and Near Misses (virtually no research yet)
  - Improvising Solutions (essential no research yet)

- Part 2B: Combined AI + Human Teams
  - Maintain Situational Awareness (past systems had very poor situational awareness)
    - Opportunity: Transparent UI to help users achieve appropriate trust
    - Need: Joint human-AI situational awareness
  - Detecting Anomalies and Near Misses (need methods that encompass the human team and other agents)
    - Opportunity: Interactive Anomaly Detection
    - Opportunity: Root Cause Analysis
  - Explaining Anomalies and Near Misses (beyond explainable AI for classifiers)
  - Improvising Solutions (AI needs to support and extend human improvisation)

Guideline for Deploying AI in High Risk Situations

- We should only apply AI in high-risk situations when we can create and maintain a high-reliability organization that combines the human and AI systems
  - AI system must support and perform the five functions of HROs

- Examples to consider:
  - Face Recognition in Law Enforcement
    - Searches against face databases vs. Real-time Body Cam identification?
  - Self-Driving Cars
    - No. A human driver can’t be an HRO. AI must be fully autonomous
  - AI Trading Systems
    - Currently lack anomaly detection, joint situational awareness. Operate too quickly for human intervention
  - Autonomous Weapons Systems
    - Maybe: Military teams already train to be HROs. But AI capabilities still lacking

QUESTIONS?

“Normal Accidents”

Charles Perrow (1984)

- Response to Three-Mile Island failures
- Claims:
  - Accidents are inevitable (“normal”) in extremely complex systems
  - If system also has catastrophic potential, these accidents will lead to catastrophe

HRO Desideratum for AI Deployment

We should not deploy AI unless we can ensure that the human organization is highly reliable